

## METHODS

# OurRank: A Software Requirements Prioritization Method Based on Qualitative Assessment and Cost-Benefit Prediction

LUIS ROJAS<sup>1</sup>, CRISTIAN OLIVARES-RODRÍGUEZ<sup>2</sup>,  
CLAUDIO ALVAREZ<sup>3</sup>, AND PEDRO G. CAMPOS<sup>4</sup>

<sup>1</sup>Departamento de Ciencias de la Computación y Tecnologías de la Información, Universidad del Bío-Bío, Chillán 3810178, Chile

<sup>2</sup>Facultad de Ciencias de la Ingeniería, Universidad Austral de Chile, Valdivia 5090000, Chile

<sup>3</sup>Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Santiago 7620086, Chile

<sup>4</sup>Departamento Sistemas de Información, Universidad del Bío-Bío, Concepción 4051381, Chile

Corresponding author: Luis Rojas (lurojas@ubiobio.cl)

**ABSTRACT** Requirements prioritization is an activity aimed at determining the essential requirements to include in a software release. Although there are several prioritization methods to systematize this task, there are still unresolved challenges. Existing methods do not guarantee that requirements prioritization meets stakeholder expectations and goals. This is because most prioritization methods operate by considering only quantitative information, making it difficult to formally capture stakeholder interests and perspectives that can rather be made explicit in qualitative terms. Likewise, methods including qualitative information only consider elements associated with benefit estimation, that is, positive aspects of the project, but neglect costs or negative aspects. As a result, the prioritization process is driven by a partial view of constraints. Such methods also fail at capturing and combining expert knowledge that decision-makers can bring into the decision-making process. In this research, we propose a novel method for software requirements prioritization, which facilitates incorporating experts' qualitative assessment at the outset of the prioritization process and considers both benefit and cost constraints. Details of the method are presented, together with a case study describing a real application scenario. Recommendations and guidelines regarding the application of the method are proposed based on the results of the case study.

**INDEX TERMS** Cost-benefit prediction, qualitative prioritization criteria, prioritization method, requirements prioritization.

## I. INTRODUCTION

Requirements prioritization is a fundamental activity in software development. It is aimed at determining requirements that are included in a software development plan considering various criteria, such as natural precedence, value generation, and risk minimization [1]. Depending on the software development process that is considered, requirements prioritization can be conducted by software engineers, requirements engineers and other specialized roles, such as product owners

The associate editor coordinating the review of this manuscript and approving it for publication was Claudia Raibulet<sup>1</sup>.

and project managers. However, despite that a wide variety of software requirement prioritization methods exist [2], these activities are often carried out in an intuitive and/or informal fashion. Dismissing the use of formal methods to conduct requirements prioritization increases the risk of missing critical issues and hinders rigorously managing consistency among requirements.

Different authors [2], [3], [4] have identified challenges for existing requirement prioritization methods, regarding efficiency and accuracy criteria. Moreover, it has been often observed that results obtained through a given method do not match priorities that are expected by decision-makers [2].

Arguably, mismatch of expectations occurs due to methods being overreliant on quantitative information [4]. Methods that only capture and process quantitative information make it difficult to reify interests and perspectives of decision-makers, which are of qualitative nature. Even though methods that include qualitative information to dynamically drive the prioritization process exist [5], [6], these only consider qualitative prioritization criteria associated with estimated benefits. Thus costs and other aspects with negative connotation fail to be considered as prioritization criteria [6]. Furthermore, cost prioritization criteria should be modelled after stakeholders' qualitative knowledge, and should be considered alongside benefit-related criteria in requirements prioritization [7]. It must be noted that other prioritization methods do not formally integrate nor combine decision-makers' prior knowledge in the prioritization process.

In this research a novel software requirements prioritization method is proposed, which, being based on previous research, seeks to overcome the limitations of current methods discussed above [6], [8], and [9]. The main contributions of this work are the following:

- A qualitative method for requirements prioritization called OurRank is presented. OurRank allows to formally capture, consider, and map benefit and cost prioritization criteria throughout the requirements prioritization process. In addition, OurRank integrates prior knowledge of decision-makers in requirement prioritization.

- A case study encompassing application and qualitative evaluation of OurRank is analyzed. The case study reports on the participation of experts in the practice of requirements prioritization by means of OurRank in a corporate setting. The results obtained permit validating the applicability and utility of the method, in relation to its intended benefits.

## A. METHODOLOGY AND STRUCTURE

This research is based on the methodology described by the Design Science framework [10], which comprises seven guidelines. Following guideline one, we have designed an innovative purposeful artifact established on a new algorithm for requirements prioritization in a specific problem domain. This, following guideline two, is the integration of qualitative prioritization criteria based on benefit and cost. Verifying guideline three, the solution proposed is suitable for the problem stated, as it is evident through the qualitative evaluation conducted, which proved to be efficient in dynamic work teams (achieving guideline four). Furthermore, as per guideline five, the proposal has been officially characterized using mathematical descriptions and internal properties, providing a search process that assembles the problem space and poses the prioritization mechanism to identify an effective solution alongside experienced decision-makers via a case study (guideline six). Finally, acknowledging guideline seven, in this paper, both the solution and results have been demonstrated to a technical and managerial audience.

This paper is organized as follows. Section II presents a revision of related work. Section III describes the formal details and operation of the proposed method. Section IV presents a case study to validate our proposal. Section V describes the results of the case study. Section VI presents a discussion of the qualitative analysis carried out on the case study. Section VII reviews the research limitations. Finally, Section VIII presents conclusions and future work.

## II. RELATED WORK

There are various prioritization methods used in software development [3], [11], [12]. Analytics Hierarchy Process (AHP) [13] identifies the priority of requirements using a pairwise comparison matrix, being one of the most influential and used methods to date [14]. Different methods based on AHP have been proposed, such as Power Analytics Hierarchy Process (PAHPT) [15] that analyzes the power relations between stakeholders, and Cost-Value [16] that prioritizes requirements based on their perceived value and implementation costs. Additionally, CBRank [17] uses machine learning to reduce the input effort while DRank [18] uses AHP and machine learning to analyze stakeholder preferences and dependencies between requirements. Finally, the Interactive Genetic Algorithm (IGA) [19] uses a genetic algorithm for incremental knowledge acquisition.

One limitation of the abovementioned methods is that they do not formally capture project priorities. Furthermore, most are based on individual assessment of requirements through peer comparisons. This prevents a global understanding of the relevant aspects to guide the prioritization process, requiring qualitative elements to capture the interests and perspectives of decision-makers.

On the other hand, the Quality Functional Deployment (QFD) method [5] is based on a matrix to represent the needs and expectations of stakeholders. The Kano method [20] also allows requirements to be prioritized based on stakeholder preferences, but focuses on differences in product features. Similarly, the Wieggers method [21] estimates the relative priorities of the requirements through a scheme based on the QFD concept of customer value. Various methods have also been proposed in the field of agile methodologies. For instance, Planning Game [22], which is part of eXtreme Programming (XP), is based on user stories to classify requirements into three priority groups; \$100 Allocation [23] or also known as Cumulative Voting, in which the stakeholders have one hundred fictitious dollars to “spend” between the requirements; Dot voting [24] that uses points assigned to each requirement; Multi Voting System [25] that allows a person to assign multiple votes on a single item; Round-the-Group Prioritization [26], in which a consensus group ordering is performed; Weighted Criteria Analysis [27] in which criteria and weights are defined for each criterion; and Binary Search tree [28] that is based on the search algorithm of the same name to compare and establish priority relationships between requirements. The MoSCoW Method [29] uses an ordinal scale of four levels that are associated with each

TABLE 1. Summary of related work.

Ref.	Method	Approach	Limitations
[13]	Analytic Hierarchy Process (AHP)	Identifies requirements priority using a pairwise comparison matrix.	These methods do not formally capture project priorities. Additionally, most are based on individual assessment of requirements through peer-reviewed comparisons. This prevents a global understanding of the relevant aspects to guide the prioritization process, requiring qualitative elements to capture the interests and perspectives of decision makers.
[15]	Power Analytic Hierarchy Process (PAHPT)	Analyzes power relations between stakeholders.	
[17]	CBRank	Uses machine learning to reduce effort in data entry.	These methods allow capturing various elements to obtain a global conception of the project (commercial objectives, customer expectations and stakeholder needs), allowing better prioritization. However, they have limitations regarding the subjective use of ordinal scales and ratings [3]. This makes it difficult to provide objective qualitative elements to clarify requirement inconsistencies during the prioritization process.
[18]	Drank	Uses AHP and machine learning to analyze stakeholder preferences and dependencies between requirements.	
[19]	Interactive Genetic Algorithm (IGA)	Uses a genetic algorithm for incremental knowledge acquisition.	These methods allow capturing various elements to obtain a global conception of the project (commercial objectives, customer expectations and stakeholder needs), allowing better prioritization. However, they have limitations regarding the subjective use of ordinal scales and ratings [3]. This makes it difficult to provide objective qualitative elements to clarify requirement inconsistencies during the prioritization process.
[5]	Quality Functional Deployment (QFD)	Uses a matrix to represent the needs and expectations of stakeholders.	
[20]	Kano	Allows prioritizing requirements based on stakeholder preferences but focuses on differences in product features.	Only considers elements associated with the estimation of benefits (positive aspects of the project), without considering its costs or negative aspects, making it difficult to fully understand the priorities of the projects.
[21]	Wieggers	Estimates the relative priorities of the requirements through a scheme based on the QFD concept of customer value.	
[6]	Qualitative Method for Prioritizing Software Requirements (QMPSR)	Considers qualitative elements related to project priorities, formalizing the assessment of requirements prioritization, and focusing on the project's relevant aspects and elements that define the priority of such requirements.	These methods focus on quantitative information.
[33]	PHandler	Relies on value-oriented prioritization and neural networks to generate cluster priority estimates, which are then evaluated via AHP.	
[34]	FBRS	Method based on optimization using fuzzy graphs and integer programming to identify value dependencies between requirements.	Focuses only on one group of requirements (security).
[39]	PAPS	Uses a fuzzy inference system to allow partial satisfaction of requirements.	
[36]	IQ-BR	Method based on quantum optimization to perform prioritization, considering interdependencies between functional and non-functional requirements.	Focuses on quantitative information.
[37]	IFA	Method based on the Intuitionist Fuzzy Approach to represent the priorities of the stakeholders, together with slicing and backtracking of requirements and page rank to analyze interdependencies between requirements.	
[9]	Quantitative RP for HCS	Quantitative method to support the prioritization of requirements in highly configurable systems (HCS).	Focuses on quantitative information, only for HCS.
[31]	DLM-MLSRP	Method that considers aspects of cost and benefit, within a pre-fixed set of aspects. The authors state the use of deep neuronal Lagrange multiplier; its application seems difficult.	
[35]	Hybrid Ranking Model	Hybrid mathematical model that combines different strategies for prioritizing requirements.	Focuses on quantitative information, does not consider negative aspects; limitations in collision handling.
[38]	LSDM-IFS	Method based on ratings by criteria associated with requirements, and the use of fuzzy intuitionist sets (Intuitionist Fuzzy Sets) to determine belonging to priority classes. Also consider using a recommendation system to provide suggestions to stakeholders and guide them to reach consensus on prioritization.	
[8]	Fuzzy Linguistic Labels	Method based on the use of fuzzy linguistic labels to establish prioritization of requirements individually by each stakeholder, to be mixed by means of the proposed MLIOWA operator. A multi-criteria prioritization method is proposed to consider different stakeholders and minimize collisions.	It is not evaluated globally (perspectives of different stakeholders are combined through a mathematical operation for aggregation of information in matrices).
[32]	SARIP	Method based on initial prioritization using the MosCoW method and assessment of a set of predefined aspects, to then use Machine Learning methods for final estimation of prioritization around three groups (low, medium and high priority) and ranking in each group.	
[29]	MosCoW	Method incorporated in the agile DSDM methodology, it is based on an ordinal scale of 4 levels that are associated with each requirement, to prioritize those of greatest value.	Limitations regarding the subjective use of ordinal scales and ratings.

requirement to prioritize those with the highest value. Similarly, the Value-Oriented method [30] focuses on elements of core value for the business. Additionally, two more recent methods include DLM-MLSRP [31] that considers aspects of cost and benefit, within a set of pre-fixed aspects, and uses the deep neural Lagrange multiplier to establish the ranking of requirements; and the SARiP method [32] that is based on an initial prioritization using the MosCoW method and the assessment of a set of predefined aspects. The latter method uses machine learning for the final estimation of prioritization around three groups (low, medium and high priority) and the corresponding ranking in each group.

These recent methods enable the capturing of various elements to obtain a global conception of the project (commercial objectives, customer expectations and stakeholder needs), allowing for better prioritization. However, they have limitations regarding the subjective use of ordinal scales and rankings [3]. This makes it difficult to provide objective qualitative elements to clarify requirements inconsistencies during the prioritization process.

Other methods mix different techniques in search of a better ranking. The PHandler method [33] generates an initial ranking of requirements by experts and then uses neural networks and the Fuzzy c-means clustering method to establish clusters of requirements according to priority, to generate a final ranking through AHP. The FBRs method [34] is based on fuzzy graph optimization and integer programming to identify value dependencies between requirements. Hybrid Ranking Model [35] contemplates a hybrid mathematical model that combines different strategies for prioritizing requirements. IQ-BR [36] is based on quantum optimization to perform prioritization, considering interdependencies between functional and non-functional requirements. Different approximations based on fuzzy logic have recently been proposed, taking advantage of its ability to represent degrees of uncertainty. For example, the Intuitionist Fuzzy Approach (IFA) [37] takes advantage of the Fuzzy method to represent the stakeholders' priorities, along with slicing and backtracking of requirements and page rank to analyze interdependencies. The LSDM-IFS method [38] uses ratings by criteria associated with requirements, and the use of intuitionist fuzzy sets to determine priority class membership. It also considers the use of a recommendation system to provide suggestions to stakeholders and guide them to reach a consensus on prioritization. However, despite the use of different techniques, these methods generally suffer from a lack of qualitative information in the prioritization process.

On the other hand, the Quantitative Requirements Prioritization method for HCS [9] uses different quantitative assessment matrices, the curriculum-based method and linear regression, to prioritize requirements in the field of Highly Configurable Systems. Similarly, the PAPS method [39] uses a fuzzy inference system to allow the partial satisfaction of requirements in the security field. These are examples of proposals focused on specific systems or types of requirements, so they are not of general applicability.

Recently proposed, the Qualitative Method for Prioritizing Software Requirements (QMPSR) [6] allows considering qualitative elements related to project priorities. However, this proposal only considers elements associated with the estimation of benefits (positive aspects of the project), without considering its costs or negative aspects, making it difficult to fully understand the priorities of the projects. Similarly, the Fuzzy method linguistics labels [8] is based on the use of fuzzy linguistic labels to establish prioritization of requirements individually by each stakeholder, considering multiple prioritization criteria that can be defined by the participants. The different preferences are then combined by a proposed new operator called MLIOWA. It also considers a multi-criteria prioritization method to consider the perspectives of the different stakeholders and minimize collisions. However, this method does not consider a global assessment in conjunction with stakeholders, which could lead to problematic collision handling. Table 1 summarizes the related work.

Considering the above mentioned, it is possible to affirm that at least part of the project' priorities are overlooked or not fully captured by the requirements prioritization methods. Most of the existing approaches are based on quantitative measures that suffer from problems in formally capturing the interests and priorities of stakeholders in the prioritization process.

### III. OURRANK: A PROPOSAL FOR SOFTWARE REQUIREMENTS PRIORITIZATION

OurRank aims to improve the prioritization process based on the incorporation of the relevant aspects of benefit (positive) and cost (negative) that define the requirements priority, via five steps, as described in Fig. 1 below. The project's relevant aspects are described and composed of a set of elements.

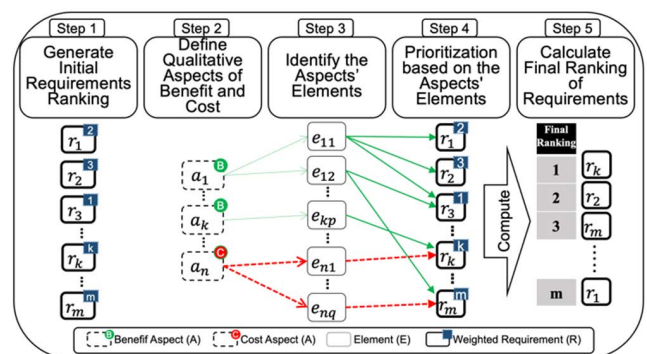


FIGURE 1. Prioritization method steps based on cost-benefit aspects.

#### A. STEP 1: GENERATE INITIAL REQUIREMENTS RANKING

The first step aims to formally capture decision-makers' prior knowledge through generation of an initial ranking requirements. This step comprises two activities. Firstly, define the initial ranking of requirements, and secondly, generate weights based on requirements' initial ranking. Details are as follows:



### 1) DEFINE REQUIREMENTS' INITIAL RANKING

This activity involves generating a first ordering or ranking of requirements, in order to capture the prior knowledge of decision-makers.

Expert knowledge is necessary to conduct a decision-making process [40]. In general, attempts are made to link and integrate knowledge with decision-making [41] through various strategies and in different contexts [42], [43], [44], [45]. That is why the prioritization process begins with the generation of an initial ordering of the requirements, to capture the prior knowledge of the decision-makers. The objective is to integrate these results in the generation of the requirements final ranking and to support situations when more than one requirement has the same priority.

### 2) GENERATE WEIGHTS BASED ON REQUIREMENTS' INITIAL RANKING

This activity consists of weighting requirements in the initial ranking, so that these considerations can be then included in final requirements assessment. Weights applied to each requirement depend on their relative rank in prioritization. Let  $R = \{r_1, \dots, r_m, \dots, r_n\}$  be a set of ordered requirements ( $r_m \in R$ ), in which  $m$  represents the order of importance within the set and  $n_r$  is the total number of requirements ( $\{n_r \in \mathbb{N} : n_r \geq 1\}$ ). The weighted value of  $r_m$ , considering its initial ranking, is defined as:

$$V(r_m) = \frac{100 - \left[ (m-1) \times \frac{(\lceil \log(n_r) \rceil \times 10)}{n_r - 1} \right]}{100} \quad (1)$$

where  $V(r_h) > V(r_m)$  implies that  $r_h (r_h \in R)$  has a higher priority than  $r_m$ .

## B. STEP 2: DEFINE QUALITATIVE ASPECTS OF BENEFIT AND COST

The second step consists of identifying the prioritization criteria that drive the process. This second step consists of three activities:

### 1) IDENTIFY PROJECT ASPECTS

Aspects of a project correspond to important issues associated with conditions that add or subtract value, according to Riegel and Doerr [7]. Some of these types include *requirement volatility or stability*, *performance*, *competitiveness*, *financial sanctions*, and *budget*, among others. This method allows the definition of a non-limited set of aspects related to benefit and cost.

### 2) DEFINE THE PRIORITY ORDER OF THE ASPECTS DEFINED FOR THE PROJECT

In this activity, the priority ordering of the previously identified aspects is determined, according to their relevance. For a set of  $n$  aspects, the least important aspect is assigned the value 1, while the most important aspect is assigned the value  $n$ .

### 3) COMPUTE ASPECTS' NORMALIZED PRIORITY VALUES

In this step, the normalized priority value of the aspects (i.e.,  $P$ ) is determined, which represents the priority with respect to the total number of aspects. A normalized value is adopted to guarantee that  $P$  will take values between  $-1$  and  $1$ , identifying whether the aspect is one of benefit or cost. Let  $A = a_1, \dots, a_k, \dots, a_n$  be a finite ordered set of aspects, where  $a_k \in A$ , such that  $k$  represents the assigned order of importance, and  $n_a$  is the total number of aspects defined  $n_a \in \mathbb{N} : n_a \geq 1$ . The normalized priority value of an aspect  $a_k$  is defined as:

$$P(a_k) = \frac{k}{n_a} BC(a_k) \quad (2)$$

where  $BC(a_k)$  identifies whether  $a_k$  is a benefit or cost aspect, defined in terms of the function  $BC$ , where  $BC: A \rightarrow \{1, -1\}$ . Thus,  $P(a_h) < 0$  identifies that aspect  $a_h$  is a cost aspect and  $P(a_h) > P(a_k)$  means that aspect  $a_h (a_h \in A)$  has a higher normalized priority than  $a_k$ .

## C. STEP 3: IDENTIFY THE ASPECTS' ELEMENTS

The third step consists of identifying and prioritizing the elements of the previously identified aspects. This step consists of two activities:

### 1) IDENTIFY AND DEFINE PROJECT ELEMENTS

Each relevant aspect is made up of a set of common elements. Elements can be divided into subcategories to identify specific elements for the relevant benefit and cost aspects [7]. For example, for the cost aspect of *Financial Sanctions*, it is possible to define elements *Costs for not implementing*, *Legal mandate*, and *Contractual commitment*, among others. This method allows the definition of an unlimited set of elements for each aspect.

### 2) PRIORITIZE ELEMENTS ASSIGNED TO EACH ASPECT

This activity consists of assigning priorities to the identified elements, considering three levels: *high*, *medium*, or *low*. This defines a total order among the elements by their priority regardless of their quantity.

Let  $E = e_{11}, \dots, e_{1z}, \dots, e_{k1}, \dots, e_{kp}, \dots, e_{n1}, \dots, e_{nq}$  be a finite set of elements, where  $z$ ,  $p$  and  $q$  correspond to the total of elements of aspects  $a_1$ ,  $a_k$  and  $a_n$ , respectively  $z, p, q \in \mathbb{N} : z, p, q \geq 1$ . The priority of an element is defined in terms of function  $L(e_{kp})$  where  $L: E \rightarrow \{1, 2, 3\}$ ; interpreting value 1 as *low* priority, 2 as *medium* priority and 3 as *high* priority.

## D. STEP 4: PRIORITIZATION BASED ON THE ASPECTS' ELEMENTS

The fourth step aims to prioritize the requirements through the aspects' elements, by identifying relationships between elements and requirements. An element can be associated with one or many requirements, and a requirement can be associated with one, many, or no elements. Such relationships

are specified and collectively negotiated between decision-makers.

The relationship between an element  $e_{kv} (e_{kv} \in E)$  and a requirement  $r_i \in R$  is defined by the function  $C(e_{kv}, r_i)$ , where  $C : E \times R \rightarrow \{0, 1\}$ ; interpreting the value 0 as no link between element and requirement and value 1 as the existence of a link between  $e_{kv}$  and  $r_i$ .

### E. STEP 5: CALCULATE FINAL RANKING OF REQUIREMENTS

The last step consists of generating the final ranking of the requirements based on the previously identified relationships and the initial ranking of the requirements. This last stage consists of two activities:

#### 1) COMPUTE IMPORTANCE OF REQUIREMENTS PER ASPECT, CONSIDERING THE INITIAL REQUIREMENTS RANKING

In this step, the importance level of the requirements is determined. The relevance of a requirement is related to the priorities of the elements of the associated aspect plus an Association Factor ( $G$ ) [6]. This  $G$  allows to prioritize elements that have more related requirements, and it can be used to compare and analyze elements with the same priority.

In this way, initially the number of requirements related to each of the elements of the project aspects is obtained. The total number of requirements related to the element  $e_{kv}$  of the aspect  $a_k$  is formally defined as:

$$TC(e_{kv}) = \sum_{i=1}^{n_r} C(e_{kv}, r_i) \quad (3)$$

where  $C(e_{kv}, r_i)$  is a function that identifies whether the element  $e_{kv}$  relates to requirement  $r_i$ . Knowing the value of  $TC$  it is possible to calculate the factor  $G$  of any element. Thus, association factor of element  $e_{kv}$  is defined as:

$$G(e_{kv}, e_{kb}) = \frac{TC(e_{kv})}{TC(e_{kb})} \quad (4)$$

where  $TC(e_{kb})$  corresponds to the maximum number of requirements to which an element of aspect  $a_x$  is linked, such that  $e_{kv} \in E, \forall e_{kx} \in E, TC(e_{kv}) \geq TC(e_{kx})$ .

With these results, it is possible to calculate the priority in the existing relationships between elements and requirements of the project. Thus, the priority of an element  $e_{kv}$  associated with requirement  $r_i$  considering its respective factor  $G$  is defined by means of function  $I: E \times R \rightarrow R$ , as follows:

$$I(e_{kv}, r_i) = \begin{cases} L(e_{kv}) + G(e_{kv}, e_{kb}), & \text{if } C(e_{kv}, r_i) = 1; \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

where  $L(e_{kv})$  defines the element's priority function and  $G(e_{kv}, e_{kb})$  corresponds to the element's association factor, for every case in which there is a link between element  $e_{kv}$  and requirement  $r_i$ , according to function  $C(e_{kv}, r_i)$ .

Once the priority of each requirement has been defined in the different elements of the project's relevant aspects, a complete assessment of the requirement is obtained, considering its initial ranking. Thus, let  $E_k = \{e_{k1}, \dots, e_{kp}\}$  be a finite sub-collection of the elements related to aspect  $a_k (E_k \subset E)$ . The total number of elements of aspect  $a_k$  assigned to requirement  $r_i$  considering its weighted value  $V$  in the initial ranking, is defined as:

$$TI(a_k, r_i) = \sum_{v=1}^{|E_k|} I(e_{kv}, r_i) V(r_i) \quad (6)$$

where  $V(r_i)$  identifies the weighted value of requirement  $r_i$  in the initial ranking, computed with the formula (1). Furthermore,  $I(e_{kv}, r_i)$  corresponds to the priority of element  $e_{kv}$  linked to requirement  $r_i$  considering its factor  $G$ , calculated according to (5).

Finally, the relevance level of requirement  $r_i$  with respect to aspect  $a_k$  as a:

$$\lambda(a_k, r_i) = \frac{TI(a_k, r_i)}{\sum_{v=1}^{|E_k|} E_k |((L(e_{kv}) + G(e_{kv}, e_{kb}))TC(e_{kv}))} 100 \quad (7)$$

where  $TI(a_k, r_i)$  defines priority of requirement  $r_i$  in all elements of aspect  $a_k$  and considering its weighted value  $V$ , as defined in equation (6). Furthermore,  $TC(e_{kv})$  identifies the total number of requirements  $r_i$  linked to element  $e_{kv}$ , calculated with (3). Hence,  $\lambda(a_k, r_i) > \lambda(a_k, r_j)$  means that  $r_i$  has greater priority than  $r_j$  for aspect  $a_k$ .

#### 2) CALCULATE THE FINAL REQUIREMENTS RANKING CONSIDERING RELEVANCE BY ASPECT

Once the importance level of the requirements for each aspect has been obtained, considering its initial ranking, the final ranking of a requirement  $r_i$ , considering its priority in all aspects, is formally defined as:

$$RFR(r_i) = \sum_{h=1}^{n_a} \lambda(a_h, r_i) P(a_h) \quad (8)$$

where  $\lambda(a_h, r_i)$  identifies the relevance level of requirement  $r_i$  with respect to aspect  $a_h$ , as defined in (7). In addition,  $P(a_h)$  defines the normalized value of cost or benefit aspect  $a_h$ , calculated by means of (2). Lastly,  $RFR(r_i) > RFR(r_j)$  means that requirement  $r_i$  has a higher priority than  $r_j$ . It must be noted that the final ranking of requirements considers all relationships amongst elements and requirements. These relationships are ought to be generated dynamically and collaboratively by stakeholders in the prioritization process. Furthermore, final priority of requirements also considers a weighted value related to the initial ranking of requirements; obtained from decision-makers' knowledge at the project outset.

In order to provide a better understanding of the phases described above, and qualitatively study how

decision-makers reason in terms of the main characteristics of our proposal, a case study comprising implementation and evaluation of the OurRank method is presented in the following section.

**IV. CASE-STUDY**

The objective of this case study was to investigate how difficult it is for decision-makers to think in terms of the main features of our proposal. Specifically, it is investigated whether the initial ranking is a suitable mechanism to capture the initial expert knowledge of decision-makers and whether the qualitative aspects is a pertinent structure to drive a software requirements prioritization process. The application scenario for OurRank in this case study was in a leading services management company in the healthcare sector. The methodology was carried out by experts from the prioritization area of the company. Research questions driving this case study are as follows:

*RQ1: Is the initial ranking an adequate mechanism to capture the expert knowledge of decision-makers and appropriate to consider it with some weighting in the final ranking?*

*RQ2: Is the structure for defining aspects, considering benefits and costs, pertinent and consistent with needs of a dynamic organization?*

**A. PARTICIPANTS**

The participants of the case study were employees of a health services management company, who usually work on tasks related to decision-making in project management. Written informed consent was obtained from all participants. A total of six engineers ( $M_{age} = 30.8$  years,  $SD_{age} = 3.4$ , ages 26 to 36; 83% male) participated in the case study. Table 2 presents characteristics of participants. All participants had experience in the application of prioritization methods and had three to ten years' experience in prioritization tasks.

**B. PROCEDURE**

The participants applied the proposed method to a practical case (see Section IV.C). They were organized into two groups ('A' and 'B', as shown in Table 2 ) to independently deal with the proposed case by fully implementing the OurRank prioritization method. Subsequently, all participants met in a focus group. The focus group was organized into three stages.

- 1) A reflection on the activities conducted, and the perceived usefulness of the method.
- 2) Information was provided to participants on the results of the tasks carried out. Based on this, participants were invited to reflect on the differences found among the results of the two groups, and possible causes of these differences.
- 3) Participants were asked to give their opinion on the advantages and disadvantages of the method.

To analyze the focus group discussions, a method based on grounded theory was followed. In grounded theory [46], analyses is derived from systematically collected data and

**TABLE 2. Characteristics of case study participants.**

#	Age	Professional Background	Occupation / Role	Requirements Prioritization Experience	Known Prioritization Methods <sup>a</sup>	Gender <sup>b</sup>	Assigned Group
P1	36	IT engineer	Project Development Chief	10	CV, PG, DV, MVS	M	A
P2	30	Industrial engineer	PMO	4	CV, PG, M, \$ADV, VO, MVS, RGP	M	A
P3	26	Industrial engineer	PM Analyst	3	CV, BST, WCA	M	A
P4	31	Industrial engineer	Principal Project Analyst	5	CV, MVS	F	B
P5	29	Industrial engineer	Project Analyst	4	CV	M	B
P6	33	IT engineer	Project Manager	6	PG, M	M	B

<sup>a</sup>BST = Binary Search Tree, CV = Cost-Value, DV = Dot Voting, M = MoSCoW, MVS = Multi-Voting System, PG = Planning Game, RGP = Round-the-Group Prioritization, WCA = Weighted Criteria Analysis, VO = Value-Oriented, \$ADV = \$100 Allocation Dot Voting.

<sup>b</sup>F = Female, M = Male

its evaluation. The process begins with description of phenomena, continues with conceptual ordering of collected data, and ends with a theorization. To analyze the content of focus group discussions, a set of theme categories was elaborated based on the interview protocol. Then this set was extended and revised as interview transcripts were coded. Based on the codes, patterns were sought to explain contexts, situations, facts and phenomena of interest.

**C. GENERAL GUIDELINES**

The application chosen to carry out the verification of our approach is an inventory, sales, and customer service system in pharmaceutical services. The end-users of this software are pharmacists, warehouse, and administrators of a certain branch of the pharmacy chain. In this way, software that has the following general guidelines is required:

- Pharmacists can create and edit a customer's profile.
- Pharmacists can check customer information.
- Pharmacists and warehouse can create and edit medication items.
- Pharmacists and warehouse can consult the movements of medicine.
- Warehouse can record stock count of medicine.
- Administrators can create and edit pharmacy branches.
- Administrators can check the information of a branch.
- Administrators can manage the profiles of both pharmacists and warehouse.

In accordance with the general guidelines described above, the requirements for this information system are presented in Table 3.

**TABLE 3. Requirements of the information system proposed for experimentation.**

R	Project Requirements
r <sub>1</sub>	Create and edit pharmacy branch
r <sub>2</sub>	Manage chain branches
r <sub>3</sub>	Check sales of a branch
r <sub>4</sub>	Integrate payment methods: debit and credit card
r <sub>5</sub>	Assign a customer service number
r <sub>6</sub>	Create and edit a customer profile
r <sub>7</sub>	View customer profile information
r <sub>8</sub>	Check a customer's purchase history
r <sub>9</sub>	Create and edit medicines
r <sub>10</sub>	Manage medications in an integrated environment
r <sub>11</sub>	Enter medication registration of a branch
r <sub>12</sub>	Enter medication withdrawal
r <sub>13</sub>	Consult drug movement
r <sub>14</sub>	Check the status of a drug
r <sub>15</sub>	Review the system as another user
r <sub>16</sub>	Provide a file repository

The general guidelines, requirements and prioritization criteria, proposed by [7], were the inputs for the participants to work collectively and collaboratively in the requirements prioritization; carrying out the different phases of the benefit-cost qualitative method presented in Section III. Specifically, the tasks were directly related to the five phases of the proposed method (see Table 4).

**TABLE 4. Description of tasks to be carried out by the participants during the experimentation.**

# Task	Task Description	# Phases of the proposed method
1	Generate a preliminary ranking of requirements	1
2	Identify prioritization aspects	2
3	Identify elements within each of the prioritization aspects	3
4	Prioritize requirements by linking aspect elements to each of the requirements	4
5	Calculate the final ranking	5

## V. RESULTS

In the following sections, the results of the implementation and complete application of the proposed method with experts from the prioritization area are presented. The results are presented according to the tasks performed by the participants (phases of the proposed method).

### A. STAGE 1: GENERATE A PRELIMINARY RANKING OF REQUIREMENTS

This first task consisted of ordering the requirements hierarchically to identify an initial ranking. Each group carried out this task collaboratively, according to their previous

knowledge and without going into details of other aspects related to the project.

The results of this task are presented in Table 5. As can be seen, the requirements  $r_9$ ,  $r_{11}$  and  $r_{12}$  have the highest priority for group A. Accordingly, these requirements have a weighted value ( $V$ ) of 1, 0.99 and 0.98, respectively, as is defined in formula (1). As for group B, the requirements  $r_{16}$ ,  $r_1$  and  $r_9$  have the highest priority. These obtain a  $V$  of 1, 0.99 and 0.98, respectively. On the contrary, the requirements  $r_2$  and  $r_{15}$  correspond to the least important for group A and B, respectively, both with a  $V$  of 0.88. It is important to note that no coincidences are identified between the initial ranking that both groups assigned to the requirements.

**TABLE 5. Preliminary ranking of stage 1 of the proposed method for groups A and B.**

R	Preliminary Ranking <sup>a</sup>		Requirements' Weighted Value ( $Vr$ )	
	Group A	Group B	Group A	Group B
r <sub>1</sub>	7	2	0.95	0.99
r <sub>2</sub>	16	5	0.88	0.97
r <sub>3</sub>	14	13	0.9	0.9
r <sub>4</sub>	10	12	0.93	0.91
r <sub>5</sub>	9	4	0.94	0.98
r <sub>6</sub>	8	7	0.94	0.95
r <sub>7</sub>	11	15	0.92	0.89
r <sub>8</sub>	12	14	0.91	0.9
r <sub>9</sub>	1	3	1	0.98
r <sub>10</sub>	4	6	0.98	0.96
r <sub>11</sub>	2	8	0.99	0.94
r <sub>12</sub>	3	9	0.98	0.94
r <sub>13</sub>	6	10	0.96	0.93
r <sub>14</sub>	5	11	0.97	0.92
r <sub>15</sub>	15	16	0.89	0.88
r <sub>16</sub>	13	1	0.9	1

<sup>a</sup>1 = highest priority, n = lowest priority.

### B. STAGE 2: IDENTIFY PRIORITIZATION ASPECTS

This task consists of identifying and prioritizing the qualitative aspects (prioritization criteria) related to the priorities of the project. In addition, the standardized priority for each relevant aspect must be calculated (see Section III.B). This task is collectively negotiated among the decision-makers according to the main guidelines and requirements of the project.

The result of this task is presented in Table 6. As can be seen, in group A, a total of 5 benefit aspects have been defined (*Benefits related to the strategy*, *Customer benefits*, *Financial benefits*, *Product or system quality*, and *Operational performance benefits*) to formally consider the related customer, product and organizational (strategic, financial and operational) priorities during the prioritization process. In addition, this group has also identified cost aspects of *Implementation risks* and *Development time costs* related to project development priorities. As for group B, the benefit aspects of *Operational performance benefits*, *Requirements*



dependency, Requirements quality, and Customer benefit have been identified, in order to take into account the priorities of organizational results and characteristics of the requirements. This group has also identified a cost aspect (Development time cost) related to project development.

On the other hand, the benefit and cost aspects in group A, of Benefits related to strategy, Customer benefits, Financial benefits, Product or system quality, Operational performance benefits, Implementation risks, and Development time costs have a priority of 7, 6, 5, 4, 3, 2, and 1, respectively, where 7 (Strategy related benefits) indicates the highest priority and 1 (Development time costs) the least important. Accordingly, group A aspects have a normalized priority (P) of 1, 0.86, 0.71, 0.57, 0.43, -0.29, and -0.14, respectively, as defined in formula (2). Note that cost aspects are identified with a negative value through the function BC. Finally, with respect to group B, the Operational performance benefits, Requirements dependency, Requirements quality, Development time cost and Customer benefit aspects have a normalized priority (P) 1, 0.80, 0.60, -0.40 and 0.20, respectively.

TABLE 6. Relevant benefit and cost aspects identified by group A and B for the information system.

Group	Aspects (A)	Aspect Type	Priority <sup>a</sup>	Priority Normalized (P <sub>n</sub> )
A	Benefits related to the strategy	Benefit	7	1
A	Customer benefits	Benefit	6	0.86
A	Financial benefits	Benefit	5	0.71
A	Product or system quality	Benefit	4	0.57
A	Operational performance benefits	Benefit	3	0.43
A	Implementation risks	Cost	2	-0.29
A	Development time costs	Cost	1	-0.14
B	Operational performance benefits	Benefit	5	1
B	Requirement dependency	Benefit	4	0.80
B	Requirements quality	Benefit	3	0.60
B	Development time cost	Cost	2	-0.40
B	Customer benefit	Benefit	1	0.20

<sup>a</sup>1 = lowest priority, n = highest priority.

C. STAGE 3: IDENTIFY ELEMENTS WITHIN EACH OF THE PRIORITIZATION ASPECTS

Once the initial ranking and the relevant aspects of the project have been identified, the next task corresponds to the identification and prioritization of the aspects' elements defined for the project in the previous stage. Table 7 presents the results of the elements identified by the participants of both groups.

Elements have been identified and prioritized to formally capture the priorities of the pharmaceutical services system,

TABLE 7. Benefit and cost elements identified for the information system's relevant aspects, in each group. Its priority calculated according to the function L is included.

Group	Aspects (A)	Aspect Type	Elements	Priority L(e <sub>ij</sub> )
A	Benefits related to the strategy	Benefit	Strategy Alignment	3
			Importance of business goals	3
			Long-term strategic value	3
A	Customer benefits	Benefit	Efficiency for the customer	2
			Competitiveness for the customer	3
A	Financial benefits	Benefit	IRR	3
A	Product or system quality	Benefit	Payment	3
			Performance	2
A	Operational performance benefits	Benefit	Sustainability	3
			Availability	3
			Scalability	3
			Accuracy	3
			Effectiveness organizational	3
A	Implementation risks	Cost	User task contribution	2
			Productivity improvement	3
			Cost reduction	1
A	Development time costs	Cost	Process efficiency	1
			Range loss	3
A	Operational performance benefits	Benefit	Restrictions organizational	2
			Product cost	3
A	Operational performance benefits	Benefit	Implementation or development cost	1
			Processes efficiency	3
B	Requirements dependency	Benefit	Operational risk reduction	2
			Products and services flow	3
B	Requirements quality	Benefit	Change management	2
			Technical and functional dependencies	3
B	Development time cost	Cost	Traceability	3
			Consistency	3
B	Customer benefit	Benefit	Integrity	3
			Product cost	3
B	Customer benefit	Benefit	Customer efficiency	3
			Customer satisfaction	2

in order to drive the software requirements prioritization process. For example, for the Operational performance benefits aspect, the element User task contribution has been defined to formally consider the importance of the requirements that provide added value to the proposed system's end-users. In this same aspect, the element Cost reduction has also been defined with a lower priority than Productivity improvement (that is, the elements have a priority L of 1 (low) and 3 (high), respectively), in order to give a higher priority to those requirements that improve efficiency of the results of the task, over those related to the effectiveness of its costs.

As shown in Table 7, the group A aspects of Benefits related to strategy, Customer benefits, Financial benefits, Product or system quality, Operational performance benefits, Implementation risks and Development time costs are made up of 3, 2, 2, 5, 5, 2, and 2 elements, respectively. Likewise, the group B aspects of Operational performance benefits, Requirements dependency, Requirements quality, Development time cost, and Customer benefit are made up of 4, 1, 3, 1, and 2 elements, respectively. In addition, each element is assigned a priority (L), using the High, Medium or

Low scale. In this way, the result of this task is the set of elements according to the relevant aspects of the proposed pharmaceutical services system.

**D. STAGE 4: PRIORITIZE REQUIREMENTS BY LINKING ASPECT ELEMENTS TO EACH OF THE REQUIREMENTS**

Once the initial ranking and the relevant aspects' elements of the pharmaceutical services system have been identified, the second last stage of the prioritization method consists of materializing the requirements prioritization. This task consists of identifying the relationships between the relevant aspects' elements (see Table 7 ) and the project requirements (see Table 3 ). Like the previous tasks, these relationships are collectively and dynamically identified and agreed upon by decision-makers (group participants).

**TABLE 8. Prioritization process based on the relationship between the aspects' elements and project's requirements. Group A.**

Aspects	Elements	Requirements															
		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
Benefits related to the strategy	Strategy	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	Alignment	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
	Importance of business goals	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
	Long-term strategic value	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
Customer benefits	Efficiency for the customer	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
	Competitiveness for the customer	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Financial benefits	IRR	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Payment	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Product or system quality	Performance	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0
	Sustainability	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
	Availability	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	0
	Scalability	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
	Accuracy	0	0	0	1	1	1	1	0	1	1	0	0	0	0	0	0
Operational performance benefits	Effectiveness organizational	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	User task contribution	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Productivity improvement	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	Cost reduction	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
Implementation on risks	Process efficiency	1	1	1	1	0	0	1	1	0	0	0	1	1	0	0	0
	Range loss	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	Restrictions organizational	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Development time costs	Product cost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Implementation or development cost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The results of this task are presented in Table 8 and Table 9 for group A and B, respectively. As can be seen, while the first and second columns represent the aspects and elements, respectively, the following columns represent the project requirements. In this way, it is possible to identify the requirements (marked 1) related to each element, as defined in function C (described in Section III.D).

**TABLE 9. Prioritization process based on the relationship between the aspects' elements and project's requirements. Group B.**

Aspects	Elements	Requirements															
		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
Operational performance benefits	Processes efficiency	0	0	0	0	1	0	0	0	1	1	1	1	1	1	0	1
	Operational risk reduction	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0
	Products and services flow Change management	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Requirement dependency	Technical and functional dependencies	0	1	1	0	0	0	1	1	0	1	1	1	1	1	0	0
	Traceability	1	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0
Requirements quality	Consistency	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	Integrity	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1
Development time cost	Product cost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Customer benefit	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Customer benefit	Customer efficiency	0	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0
	Customer satisfaction	0	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0

**E. STAGE 5: CALCULATE THE FINAL RANKING**

Once the relationships between the project requirements and relevant aspects' elements have been identified, the last stage of the proposed method consists of calculating the requirements final ranking (RFR). To carry out this task, the method considers the different existing relationships (that is, the results of Table 8 and Table 9 for group A and B, respectively) and the functions described in formulas 3 to 8 (Section III.E).

Specifically, to calculate the RFR, two subtasks are performed. The first subtask consists of calculating the relevance level that the requirements have in each of the defined aspects, considering at the same time their initial ranking. First, the total number of related requirements in each element is computed through the function TC, defined in formula (3). For example, in group A, the elements Availability and Accuracy of the Product or system quality aspect have seven related requirements (see Table 8 ). These elements have the maximum number of requirements to which an element of the Product or system quality aspect has been associated. Therefore, the Association Factor (G) for both elements is 1, according to formula (4). Once obtained from the value of G, it is possible to calculate the priority of a certain requirement in some of the related elements. Continuing with the same example, requirement r5 has in the elements Scalability and Accuracy of the Product or system quality aspect a priority I, defined in formula (5), of 3.57 and 4, respectively. It should be noted that the priority I is different even when both elements have the same priority L (high) since their association factor G is different (that is, 0.57, and 1 for the elements Scalability and Accuracy, respectively). However, the priority above only

considers the individual evaluations of each element in the requirements, without a total estimate. That is why formulas (6) and (7) are computed to generate a relevance level for each aspect, considering the initial ranking of the requirements. Table 10 shows the relevance level of each requirement (that is, from  $r_1$  to  $r_{16}$ ) regarding *Product or system quality* aspect, as a result of applying the relevance level of the requirements by aspect  $\lambda$ . In this way, the results for the requirements  $r_1$  (4.66%) and  $r_2$  (4.32%) indicate that  $r_1$  has higher priority than  $r_2$  for the *Product or system quality* aspect. Finally, in the second subtask, the requirements final ranking is generated considering the relevance level of each requirement per aspect and its normalized priority ( $P$ ), calculated with formula (8).

**TABLE 10.** Relevance level of the requirements regarding Product or system quality aspect.

Aspects (A)	Requirements Relevance Level in Percentage Terms ( $\lambda$ )															
	$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	$r_7$	$r_8$	$r_9$	$r_{10}$	$r_{11}$	$r_{12}$	$r_{13}$	$r_{14}$	$r_{15}$	$r_{16}$
Product or system quality	4.66	4.31	7.02	8.36	8.43	3.14	8.83	8.75	9.60	4.40	9.52	9.45	5.27	5.31	2.96	0

Table 11 and Table 12 present the requirements final ranking of the proposed pharmaceutical services system for groups A and B, respectively. As can be seen, the requirements are ordered according to their final ranking, calculated with formula (8). Thus, in group A  $r_{10}$ ,  $r_7$  and  $r_{12}$  are the highest priority requirements, obtaining a final ranking of 75.92, 56.32 and 42.76, respectively. In group B,  $r_{10}$ ,  $r_2$  and  $r_{11}$  have the highest priority, that is, a final ranking of 24.66, 22.05 and 19.33, respectively. It should be noted that both groups have defined the requirements  $r_{10}$  and  $r_{16}$  with greater and lesser importance, respectively.

In short, the application of this case study using the prioritization method in both groups has allowed the participants to fully experience the different characteristics of our proposal. In the following section, a qualitative analysis is provided, obtained from the results of the focus group carried out with all the participants, to understand how the decision-makers reason with respect to the main characteristics of the method.

**VI. DISCUSSION**

The discussion is presented in the order of the research questions described above. As usual, in qualitative studies, we will complement our findings with quotes from the participants.

*RQ1: Is the initial ranking an adequate mechanism to capture the expert knowledge of decision-makers and appropriate to consider it with some weighting in the final ranking?*

To define the initial ordering, a preliminary analysis of the requirements by the team members is performed. Each participant provides both quantitative and qualitative arguments based on their own perceptions. To reach a group decision, every participant’s opinion regarding their prioritization is highlighted. Therefore, their previous experiences and knowledge relate to the current decision process.

**TABLE 11.** Requirements final ranking in group A, ordered by their importance descending.

R	Project Requirements	Final Ranking (FR)
		Group A
$r_{10}$	Manage medications in an integrated environment	75.92
$r_7$	View customer profile information	56.32
$r_2$	Manage chain branches	42.76
$r_8$	Check a customer’s purchase history	30.94
$r_5$	Assign a customer service number	30.86
$r_{11}$	Enter medication registration of a branch	12.61
$r_{12}$	Enter medication withdrawal	12.51
$r_9$	Create and edit medicines	11.98
$r_4$	Integrate payment methods: debit and credit card	11.38
$r_{14}$	Check the status of a drug	10.64
$r_{13}$	Consult drug movement	10.55
$r_3$	Check sales of a branch	10.39
$r_6$	Create and edit a customer profile	7.93
$r_1$	Create and edit pharmacy branch	-2.21
$r_{15}$	Review the system as another user	-3.40
$r_{16}$	Provide a file repository	-5.18

**TABLE 12.** Requirements final ranking in group B, ordered by their importance descending.

R	Project Requirements	Final Ranking (FR)
		Group B
$r_{10}$	Manage medications in an integrated environment	24.66
$r_2$	Manage chain branches	22.05
$r_{11}$	Enter medication registration of a branch	19.33
$r_8$	Check a customer’s purchase history	19.29
$r_{12}$	Enter medication withdrawal	19.16
$r_{13}$	Consult drug movement	16.60
$r_{14}$	Check the status of a drug	16.46
$r_5$	Assign a customer service number	14.91
$r_7$	View customer profile information	13.71
$r_1$	Create and edit pharmacy branch	13.08
$r_{15}$	Review the system as another user	12.47
$r_3$	Check sales of a branch	11.62
$r_9$	Create and edit medicines	6.08
$r_6$	Create and edit a customer profile	5.53
$r_4$	Integrate payment methods: debit and credit card	2.68
$r_{16}$	Provide a file repository	2.38

Next, the team must agree on the preliminary ordering that will be proposed as the first phase of the method. This process supports the development of an essential element: negotiation. This dynamic within the group allows the participants to establish their positions and interests. The decision

process is a prioritization problem based on alternatives that are beneficial to all stakeholders. The preliminary ordering is, then, the product of an exchange of personal positions based on previous experiences. Therefore, this initial ordering becomes the base of the project's aspects and elements for the second phase of the prioritization method.

The prioritization is oriented towards an enriched negotiation of the participants through an analysis of relevant aspects and elements from professional experience. Meanwhile, the methods of the literature are aimed at reducing the complexity of the discussion by defining objectives or pre-established criteria [11], incorporating the power relationships of the stakeholders [13], or through the automatic discovery of contexts from the documentation [15], [16], [17]. Therefore, this new approach is aligned with the new team structure methods by a more horizontal and people-centered way, formally capturing project priorities and team characteristics. This expert knowledge base becomes explicit by means of the proposed method and evolves to be expressive and open. This, provides sufficient evidence to apply machine learning methods in order to discover inherent relationships between aspects, like the proposals of CBRank [15] and DRank [16].

The negotiation of the project's relevant aspects and elements is the key item for discovering the priorities. Thus, revealing the particularities of each project, by focusing on the benefits for the users and the business, or on the technical complexities of the product properties. This increases the qualitative expressiveness of the proposed method compared to those that focus on the needs and expectations of the stakeholders [5], [18], [19].

This qualitative expressiveness is reflected in the ability to explain aspects that influence the final ranking, which emerge from the expert knowledge of the members of the prioritization team.

*RQ2: Is the structure for defining aspects, considering benefits and costs, pertinent and consistent with needs of a dynamic organization?*

During the focus group session, developed with both groups, some criteria used during the preparation of the initial ranking emerged. One of the first criteria considered in Group B, among the requirements presented, is oriented to the development dependencies, where the participants mention that they have sought to give technical coherence to the requirements:

*"...trying to give it an order, a logical order as to how, for example, if I want to manage something, first I have to create it."*

Meanwhile, within the same group, when assigning a priority to the requirements, criteria aimed at business management were considered in combination with dependence on development, as the participants indicated that:

*"We didn't give dependency as much priority either, being that we look at the preliminary to order it, that's why there is a mismatch"*

By contrary, group A proposes a strategy more focused on the business, stating that:

*"Per se, prioritize one more business area or an area like this"*

Therefore, business or development-oriented profiles are outlined in the selection of criteria for the justification and assignment of the initial ranking in this working group.

It is important to consider that the technical dependencies between the requirements are never set aside when evaluating coherence by group A, since:

*"When one prioritizes activities as such of a project, one first has the technical restrictions of saying, I need this and this because. if not, it has no coherence"*

Neither are the relationships with the business goals by group B dismissed, which is reflected in the prioritization through criteria associated with the observable benefit for the client:

*"We gave a high score to the benefit to the client during the prioritization exercise that we did"*

Subsequently, both groups considered business and technical elements when preparing the initial prioritization of requirements. However, the more relevant aspects in the prioritization are also evident; business focused by group A and development focused by group B. All this is consolidated in the initial prioritization of the requirements that is observed in Table 5. Group A gives the highest priorities to the requirements that deliver the greatest benefits to the business, that is, *Create and edit medications*, *Enter medication registration of a branch*, and *Enter medication withdrawal*. Meanwhile, group B gives a higher priority to the requirements based on their technical dependency, which are, *Provide a file repository*, *Create and edit pharmacy branch*, and *Create and edit medicines*.

The negotiation not only contributes to the explanation of the participants' personal positions, but also establishes the team profile. The ranking is the solution that satisfies most of the parties involved in the analysis; the requirements at the top positions outline the focus of the entire team. Group A, for example, has requirements oriented to the client and the impact they have on critical business processes. Meanwhile, group B oriented their decisions to technical aspects that would allow an incremental development of the software product. Therefore, the establishment of the project's aspects and elements in the prioritization method strongly depends on the negotiation of the preliminary ranking. This is clearly reflected in the aspects of both groups (Table 7), where group A opted for business benefits, such as *Benefits related to the strategy*, *Customer benefits*, and *Financial benefits* and group B preferred operational performance aspects and requirements dependencies, such as *Operational performance benefits*, *Requirements dependency*, and *Requirements quality*. Therefore, in both cases, the relationship between the prioritization positions and the prioritized aspects is direct and consistent with the profile of each group.

Consequently, expert knowledge is strongly represented during the analysis of the preliminary ranking. Therefore, such knowledge is integrated into the final ranking through aspects based on personal positions during negotiation.



The qualitative characteristics, based on the expert knowledge, are clear in the definition of cost and benefit aspects in a dynamic and agile way. Both the project and organizational level are sources of knowledge to analyze the aspects. One of the participants indicated that *“a project manager can actually prioritize their tasks internally, but for an organization it is much more valuable when you are already sorting by project level”*, recognizing intrinsic business restrictions that influence ordering decisions and, consequently, the aspects and elements. On the other hand, the business strategy includes the software development process. This is reflected by the participants’ discussions during the focus group session, as they gave *“a high score to the benefit to the client because of the prioritization exercise that we did [in the company]”*. The participants also considered the impact that the requirements will have on the organizational needs, principally the priority they may have for the client’s business as *“what we pay most attention to is the impact on the client”*. These organizational needs are strongly consistent with the definition of the aspects, since the *Customer benefit* was recognized by both team A and B (6 (high) and 1 (low), respectively). Thus, the proposed method can define team profiles.

Consequently, the aspects definition is constrained by two sources: 1) the business, where the aspects associated with benefits and impact on the business are prominent; and 2) the project, where the technical tasks emerge to support the aspects. During the study, two analysis strategies were observed. While the first group prefers the benefits; strategy, clients, and finances; the second group favors operational and technical aspects associated with the requirements, as can be seen in Table 7. Therefore, since both teams belonged to the same organization, aspects that cover the business strategy develop. However, the professional profile of team members supports the generation of aspects and recognizes the participants’ previous experience. Likewise, the teams manage to identify aspects not only associated with benefits, but also with project and business costs, which are relevant and pertinent for an appropriate adjustment to the requirements prioritization method.

## VII. RESEARCH LIMITATIONS AND THREATS TO VALIDITY

The method proposed can be put into practice by teams in professional or research contexts, according to definitions in each of its stages. The results of the case study described, in which two different professional teams participated, show that the method by itself does not guarantee that its results can be consistently reproduced by different teams. That is to say, two different teams may prioritize requirements differently when applying the method. Moreover, the results obtained by either of the two teams analyzed in this study are not necessarily results that would satisfy expectations and preferences of every competent decision-maker in the organization.

The foregoing is not a limitation that exclusively affects the method presented, but it is a limitation that can potentially affect any method that includes decision-making based on evaluations or judgments by domain experts in an organization. According to the literature, we identify that this can be influenced by different factors. Firstly, it must be acknowledged that requirements engineering activities are knowledge-intensive [47], [48], and that characteristics of knowledge management processes in the organization will significantly influence results of requirements prioritization [49], [50]. Secondly, social interactions that occur among experts who enact the method, and the conditions that facilitate or hinder these interactions, such as cultural factors and organizational norms and structures, will shape the way prioritization decisions are made [51], [52]. Also, the organization’s vision and the ability of its leaders to synchronize middle managers and executive roles at different levels with it can influence the requirements prioritization that is achieved through the proposed method [53], [54], [55]. Lastly, the proposed requirements prioritization method falls within the category of group decision making processes [56]. In these processes it is common that decision makers elicit assessments quantitatively. However, it is usually difficult for decision makers to elicit quantitative assessments on aspects that cannot be compared quantitatively and objectively in a straightforward manner. Thus, prioritization can be indeed influenced by domain experts’ subjectivity, to the extent that it can be even influenced by their own cognitive distortions, such as the sunken costs fallacy, or confirmation bias. In the case study here presented, this can be considered a threat to validity of results.

Regarding knowledge management in the requirements prioritization process, it must be considered that expert knowledge in the organization takes two forms; tacit and/or explicit [54]. Tacit knowledge is difficult to communicate, transfer and share, as it is highly personal, whereas explicit knowledge can be better communicated, internalized, and synthesized by the parties that collaborate in a requirements prioritization process. Reliance on tacit knowledge, without sufficient explanation and systematization, that is, through adequate documentation and artifacts that allow collaborators to obtain all the relevant information for decision-making, is one of the reasons why different teams may prioritize requirements differently [47], [48]. Therefore, inconsistent results cannot be attributed to the requirements prioritization method itself, but rather to the knowledge management processes that occur in the context where the method is applied. Certainly, the method presented here can be accompanied by practices, tools, artifacts, and knowledge assets that facilitate the availability of complete and consistent explicit knowledge to collaborators. However, this is an aspect beyond the scope of the proposal hereby presented. In addition, the presented case study did not incorporate formal knowledge management methods to deal with these issues, which can be considered a threat to validity of results, due to omission of relevant information or project details. Thus further research

is required to determine facilitators of the knowledge management process and the experts who determine the initial requirements prioritization.

In organizational cultures where there are highly hierarchical power relations, conformity with well-established social norms, and conflict avoidance by people, decision makers with higher professional status and greater experience will likely steer discussions and ultimately dominate the decision-making process in favor of their position [51], [52]. However, for the requirements prioritization methodology described here, and its success at meeting strategic objectives and stakeholder expectations, it is desirable that the prioritization process collects judgments of various collaborators, such as managers, executives, experts, and stakeholders from different functional areas, and even stakeholders beyond organizational boundaries to reconcile their interests. In organizations where rivalry or competitiveness exists among different departments or teams, or where a team enjoys greater social reputation or valuation over another, sources of bias may arise in the prioritization process due to power struggles or conflicting interests. This will also mean that the results will not be independently reproducible by different teams in the same organization. It must be noted that in the case study presented here there was no prior evaluation of existing corporate culture to control for these issues, thus this omission can be considered a threat to validity of results. The potential biases identified here could be moderated through the use of technologies for decision-making in a collaborative context, including Group Decision Support Systems (GDSS) [56] that, for example, support the prioritization process through anonymous voting by collaborators, and facilitate negotiation to reach consensual agreements [57].

Finally, an organization can be understood as a space in which knowledge is created [54]. Therefore, decisions made have an inextricable relationship with the knowledge that is created. In the organization, middle managers act as coordinators and executors of the knowledge generation process. Leaders, on the other hand, provide corporate the vision and synchronize the organization with it. Vision determines the identity of the organization, its purpose, its direction and how it achieves its objectives [55]. Also, vision determines the value system that evaluates, justifies and determines the quality of knowledge that informs decisions. For this reason, the requirements engineering processes, and in particular, the implementation of the method proposed in this research, must be carried out under the vision transmitted by the leadership in the organization. If in a software requirements prioritization process focus is on different types of users and their experiences and needs, on process innovation to achieve greater efficiencies, or on cost minimization for competitiveness, this must be understood by decision-makers from a shared vision in the requirements prioritization process. A disparity in relation to this understanding could result in disconnection with the objectives or the vision that should prevail in the organization [53].

## VIII. CONCLUSION AND FUTURE WORK

This work presents a qualitative method to prioritize software requirements, considering aspects and elements of benefits and costs, which define the requirements' relevance. The method aims to lead the prioritization process by using qualitative elements to formally incorporate different perspectives based on the professional experiences of each of the team members, supporting the prioritization decisions. The formulation of the method has been presented, as well as the validation through a real use case of the method, allowing analyze of the main characteristics and highlighting the benefits of the qualitative prioritization process based on negotiation and dialogue, key principles in today's software development processes - mainly in people-focused agile developments.

The proposed method is aligned with agile development values that are commonly pursued nowadays in the software industry. In particular, the method provides a well-defined process to achieve effective prioritization. However, stakeholder's qualitative insights are taken into account at the outset of the process, thus it can be affirmed that in the context of the proposed method people are indeed valued over processes. This is an innovative feature of the methodology, when compared to other current prioritization methods found in the literature [6], [8], [9]. The case study presented validates that, by means of the method, a productive social context is elicited within the organization as stakeholders discuss and collaborate on relevant aspects and elements that underpin prioritization decisions.

This people-focused aspect of the proposed method provides a safe space for dialogue between team members, one of the main principles of modern software companies. It is this dialogue, present in the method and in current organizations, that enables negotiations to occur which highlight the professional experience of everyone, that is, expert knowledge that positively influences the considerations generated by the team. The prioritization dynamic contributes to outlining the final ranking based on the professional characteristics of the team. In other words, from the case study it is possible to identify the existence of prioritizations based on the product and the business, which depend on the expert knowledge of the team.

Likewise, the dynamic discussions, proposed by the method, support the definition of the aspects and elements that arise from the experience of the team's professionals. This consolidates a shared knowledge base among the members of each team in terms of evaluation criteria for project requirements. This is not only useful for the ongoing evaluation, but also consolidates and clarifies the knowledge as ranking, aspects or elements that has been generated by the team.

Consequently, the above facilitates nurturing a knowledge base of aspects that can benefit or limit software projects, allowing to appropriately adjust the methodological strategies of software development. This knowledge base allows to drive the change, or the probability of change, in the software

businesses' industry, and can support developing automated text analysis methods, allowing implicit relationships to be extracted in the aspects present in the team's knowledge base and, with this, determine regularities specific to each team in the prioritization processes.

As for future work, we hope to extend this approach with methods to deal with multi-criteria group decision making. Linguistic decision-making models that apply computing with words present themselves as a promising approach for dealing with decision problems with high uncertainty and impreciseness. Also, the proposed approach can benefit from adopting a formal language to capture the description of the benefit and cost aspects, which allows consolidating an expert knowledge base shared by software development communities. Likewise, we hope to study the application of the proposed method in contexts of use other than software development, in order to know in detail, the specific objectives that decision makers want to achieve. Similarly, we will tackle the task of discovering new prioritization criteria, which are efficiently adapted to each specific use context. We also hope to carry out long-term comparative experiments, with synthetic data and real users, to evaluate specific characteristics of the proposal. In addition, we want to incorporate various deep learning techniques that allow generating automatic recommendations of relevant elements and aspects according to the main characteristics and guidelines of the projects. Also, we hope to study and identify the different facilitators of the expert knowledge management process, to support the tasks related to the initial requirements prioritization. Finally, we hope to build an easy-to-use CASE tool to apply the proposal and make its use more widespread in modern software companies.

## REFERENCES

- [1] K. Pohl, *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam-Foundation Level-IREB Compliant*. San Rafael, CA, USA: Rocky Nook, Inc., 2016.
- [2] K. Curcio, T. Navarro, A. Malucelli, and S. Reinehr, "Requirements engineering: A systematic mapping study in agile software development," *J. Syst. Softw.*, vol. 139, pp. 32–50, May 2018.
- [3] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, "A systematic literature review of software requirements prioritization research," *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 568–585, 2014.
- [4] M. I. Babar, M. Ramzan, and S. A. K. Ghayyur, "Challenges and future trends in software requirements prioritization," in *Proc. Int. Conf. Comput. Netw. Inf. Technol.*, Jul. 2011, pp. 319–324.
- [5] F. Franceschini, *Advanced Quality Function Deployment*. Boca Raton, FL, USA: CRC Press, 2016.
- [6] L. A. Rojas and J. A. Macías, "Toward collisions produced in requirements rankings: A qualitative approach and experimental study," *J. Syst. Softw.*, vol. 158, Dec. 2019, Art. no. 110417.
- [7] N. Riegel and J. Doerr, "A systematic literature review of requirements prioritization criteria," in *Proc. Int. Work. Conf. Requirements Eng., Found. Softw. Quality*, 2015, pp. 300–317.
- [8] G. D. Rottoli and C. Casanova, "Multi-criteria and multi-expert requirement prioritization using fuzzy linguistic labels," *ParadigmPlus*, vol. 3, no. 1, pp. 1–18, Feb. 2022.
- [9] A. Ali, Y. Hafeez, S. Hussain, and S. Yang, "Role of requirement prioritization technique to improve the quality of highly-configurable systems," *IEEE Access*, vol. 8, pp. 27549–27573, 2020.
- [10] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quart.*, vol. 28, no. 1, pp. 75–105, Mar. 2004.
- [11] E. M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile requirements engineering: A systematic literature review," *Comput. Stand. Interface*, vol. 49, pp. 79–91, Jan. 2017.
- [12] S. U. Muneer, M. Nadeem, and B. Kasi, "Comparison of modern techniques for analyzing NFRs in Agile: A systematic literature review," *J. Softw. Eng. Pract.*, vol. 3, no. 3, pp. 1–12, 2019.
- [13] T. L. Saaty, "Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process," *Revista Real Academia Ciencias Exactas, Fisicas Naturales, A, Matematicas*, vol. 102, no. 2, pp. 251–318, Sep. 2008.
- [14] F. A. Bukhsh, Z. A. Bukhsh, and M. Daneva, "A systematic literature review on requirement prioritization techniques and their empirical evaluation," *Comput. Standards Interfaces*, vol. 69, Mar. 2020, Art. no. 103389, doi: 10.1016/J.CSL.2019.103389.
- [15] O. Ibrahim and A. Nosseir, "A combined AHP and source of power schemes for prioritising requirements applied on a human resources," in *Proc. MATEC Web Conf.*, vol. 76, 2016, p. 04016.
- [16] J. Karlsson and K. Ryan, "A cost-value approach for prioritizing requirements," *IEEE Softw.*, vol. 14, no. 5, pp. 67–74, Sep./Oct. 1997.
- [17] A. Perini, A. Susi, and P. Avesani, "A machine learning approach to software requirements prioritization," *IEEE Trans. Softw. Eng.*, vol. 39, no. 4, pp. 445–461, Apr. 2013.
- [18] F. Shao, R. Peng, H. Lai, and B. Wang, "DRank: A semi-automated requirements prioritization method based on preferences and dependencies," *J. Syst. Softw.*, vol. 126, pp. 141–156, Apr. 2017.
- [19] P. Tonella, A. Susi, and F. Palma, "Interactive requirements prioritization using a genetic algorithm," *Inf. Softw. Technol.*, vol. 55, no. 1, pp. 173–187, Jan. 2013.
- [20] M. Cantu, J. Kim, and X. Zhang, "Finding hash collisions using MPI on HPC clusters," in *Proc. IEEE Long Island Syst., Appl. Technol. Conf. (LISAT)*, May 2017, pp. 1–6.
- [21] K. Wiegiers and J. Beatty, *Software Requirements*. London, U.K.: Pearson, 2013.
- [22] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004.
- [23] D. Leffingwell and D. Widrig, *Managing Software Requirements: A Use Case Approach*. London, U.K.: Pearson Education, 2003.
- [24] E. Gottesdiener, "At a glance: Other prioritization methods," EBG Consulting Inc., Tech. Rep., 2007. [Online]. Available: <http://www.ebgconsulting.com>
- [25] J. Tabaka, *Collaboration Explained: Facilitation Skills for Software Project Leaders*. London, U.K.: Pearson, 2006.
- [26] Z. Bakalova, M. Daneva, A. Herrmann, and R. Wieringa, "Agile requirements prioritization: What happens in practice and what is described in literature," in *Proc. Int. Work. Conf. Requirements Eng., Found. Softw. Quality*, 2011, pp. 181–195.
- [27] Z. Racheva, M. Daneva, A. Herrmann, and R. J. Wieringa, "A conceptual model and process for client-driven agile requirements prioritization," in *Proc. 4th Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, May 2010, pp. 287–298.
- [28] V. Ahl, "An experimental comparison of five prioritization methods: Investigating ease of use, accuracy and scalability," M.S. thesis, School Eng., Blekinge Inst. Technol., Ronneby, Sweden, 2005.
- [29] D. Tudor and G. A. Walter, "Using an agile approach in a large, traditional organization," in *Proc. AGILE*, 2006, p. 373.
- [30] J. Azar, R. Smith, and D. Cordes, "Value-oriented requirements prioritization in a small development organization," *IEEE Softw.*, vol. 24, no. 1, pp. 32–37, Jan. 2007.
- [31] R. Devadas and N. G. Cholli, "Multi aspects based requirements prioritization for large scale software using deep neural Lagrange multiplier," in *Proc. Int. Conf. Smart Technol. Syst. Next Gener. Comput. (ICSTSN)*, Mar. 2022, pp. 1–6.
- [32] F.-F. Chua, T.-Y. Lim, B. Tajuddin, and A. P. Yanuarifiani, "Incorporating semi-automated approach for effective software requirements prioritization: A framework design," *J. Informat. Web Eng.*, vol. 1, no. 1, pp. 1–15, Mar. 2022.
- [33] M. I. Babar, M. Ghazali, D. N. A. Jawawi, S. M. Shamsuddin, and N. Ibrahim, "PHandler: An expert system for a scalable software requirements prioritization process," *Knowl.-Based Syst.*, vol. 84, pp. 179–202, Aug. 2015.
- [34] D. Mougouei, A. Ghose, H. Dam, M. Fahmideh, and D. Powers, "A fuzzy-based requirement selection method for considering value dependencies in software release planning," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Jul. 2021, pp. 1–6.



- [35] I. P. Gambo, R. Ikono, O. G. Iroju, T. O. Omodunbi, and O. K. Zohoun, "Hybridized ranking model for prioritizing functional software requirements: Case study approach," *Int. J. Softw. Innov. Int. J. Swarm Intell.*, vol. 9, no. 4, pp. 1–31, 2021.
- [36] R. Devadas and N. G. Chollu, "Interdependency aware QUBIT and BROWNBOOST rank for large scale requirement prioritization," *Int. J. Comput. Digit. Syst.*, vol. 11, no. 1, pp. 625–634, 2022.
- [37] A. Gupta and C. Gupta, "A novel collaborative requirement prioritization approach to handle priority vagueness and inter-relationships," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 5, pp. 2288–2297, May 2022.
- [38] V. C. Gerogiannis and D. Tzimos, "Intuitionistic fuzzy sets in large-scale software requirement prioritization," in *Handbook of Research on Advances and Applications of Fuzzy Sets and Logic*. Hershey, PA, USA: IGI Global, 2022, pp. 443–476.
- [39] D. Mougouei, D. M. W. Powers, and E. Mougouei, "A fuzzy framework for prioritization and partial selection of security requirements in software projects," *J. Intell. Fuzzy Syst.*, vol. 37, no. 2, pp. 2671–2686, Sep. 2019.
- [40] R. Grundmann, "The problem of expertise in knowledge societies," *Minerva*, vol. 55, no. 1, pp. 25–48, Mar. 2017.
- [41] A. Pagano, R. Giordano, and M. Vurro, "A decision support system based on AHP for ranking strategies to manage emergencies on drinking water supply systems," *Water Resour. Manage.*, vol. 35, no. 2, pp. 613–628, Jan. 2021.
- [42] C. Gangl, M. Lackner, J. Maly, and S. Woltran, "Aggregating expert opinions in support of medical diagnostic decision-making," in *Knowledge Representation for Health Care, ProHealth (KR4HC)*, 2019, pp. 56–62.
- [43] S. Kareksela, K. Aapala, A. Alanen, T. Haapalehto, J. S. Kotiaho, J. Lehtomäki, N. Leikola, N. Mikkonen, A. Moilanen, E. Nieminen, S. Tuominen, and R. Virkkala, "Combining spatial prioritization and expert knowledge facilitates effectiveness of large-scale mire protection process in Finland," *Biol. Conservation*, vol. 241, Jan. 2020, Art. no. 108324.
- [44] S. L. Philpot, K. W. Hipel, and P. A. Johnson, "Design and application of a values-focused decision support system to facilitate public participation in environmental decision-making," *Hohenheim Discuss. Papers Bus., Econ. Social Sci.*, vol. 2017, p. 163, Aug. 2017.
- [45] H. Zhang and D. W. R. Marsh, "Generic Bayesian network models for making maintenance decisions from available data and expert knowledge," *Proc. Inst. Mech. Eng., O. J. Risk Rel.*, vol. 232, no. 5, pp. 505–523, Oct. 2018.
- [46] K. Charmaz, "Grounded theory in the 21st century: A qualitative method for advancing social justice research," in *The Handbook of Qualitative Research*, N. K. Denzin and Y. S. Lincoln, Eds. Newbury Park, CA, USA: Sage, 2005, pp. 507–535.
- [47] M. Levy, I. Hadar, and I. Aviv, "A requirements engineering methodology for knowledge management solutions: Integrating technical and social aspects," *Requirements Eng.*, vol. 24, no. 4, pp. 503–521, Dec. 2019.
- [48] K. Olmos and J. Rodas, "KMOS-RE: Knowledge management on a strategy to requirements engineering," *Requirements Eng.*, vol. 19, no. 4, pp. 421–440, Nov. 2014.
- [49] L. Pilat and H. Kaindl, "A knowledge management perspective of requirements engineering," in *Proc. 5th Int. Conf. Res. Challenges Inf. Sci.*, May 2011, pp. 1–12.
- [50] E. Serna M., O. Bachiller S., and A. Serna A., "Knowledge meaning and management in requirements engineering," *Int. J. Inf. Manage.*, vol. 37, no. 3, pp. 155–161, Jun. 2017.
- [51] T. Alsanoosy, M. Spichkova, and J. Harland, "Cultural influence on requirements engineering activities: A systematic literature review and analysis," *Requirements Eng.*, vol. 25, no. 3, pp. 339–362, Sep. 2020.
- [52] T.-F. Kummer, J. M. Leimeister, and M. Bick, "On the importance of national culture for the design of information systems," *Bus. Inf. Syst. Eng.*, vol. 4, no. 6, pp. 317–330, Dec. 2012.
- [53] D. Callele, E. Neufeld, and K. Schneider, "Requirements engineering and the creative process in the video game industry," in *Proc. 13th IEEE Int. Conf. Requirements Eng. (RE)*, Aug. 2005, pp. 240–250.
- [54] I. Nonaka, R. Toyama, and N. Konno, "SECI, BA and leadership: A unified model of dynamic knowledge creation," *Long Range Planning*, vol. 33, no. 1, pp. 5–34, Feb. 2000.
- [55] A. Ullah and R. Lai, "Modeling business goal for business/fit alignment using requirements engineering," *J. Comput. Inf. Syst.*, vol. 51, no. 3, pp. 21–28, 2011.
- [56] J. Carneiro, P. Alves, G. Marreiros, and P. Novais, "Group decision support systems for current times: Overcoming the challenges of dispersed group decision-making," *Neurocomputing*, vol. 423, pp. 735–746, Jan. 2021.
- [57] W. Wang and M. Reani, "The rise of mobile computing for group decision support systems: A comparative evaluation of mobile and desktop," *Int. J. Hum.-Comput. Stud.*, vol. 104, pp. 16–35, Aug. 2017.



**LUIS ROJAS** received the Ph.D. degree in computer science from the Universidad Autónoma de Madrid, in 2017. He has been a Software Engineer at Chilean Nuclear Energy Commission for ten years. He is currently a Lecturer with the Universidad del Bío-Bío. His research interests include requirements engineering, empirical software engineering, and agile software development. He has also served as a reviewer at different conferences on Software Engineering and Human-Computer Interaction.



**CRISTIAN OLIVARES-RODRÍGUEZ** received the M.S. degree in computer vision and artificial intelligence from the Autonomous University of Barcelona, Barcelona, Spain, in 2010, and the Ph.D. degree in engineering from the University of Deusto. He is currently a Computer Scientist at the Catholic University of Concepción, Concepción, Chile, in 2006. He is also a Professor with the Instituto de Informática, Universidad Austral de Chile, and a Researcher at the Center for Fire and Socio-Ecological Systems (<https://www.fireses.cl/>) and the Laboratorio de Paisaje y Resiliencia Urbana (<https://pru-lab.cl/>). His research interests include modeling human behavior by means of data science, with applications in education and in heuristic problems.



**CLAUDIO ALVAREZ** received the Doctorate degree in engineering science from the Pontificia Universidad Católica de Chile. He is currently an Associate Professor in computer science at the Universidad de los Andes, Chile. His research interests include the fields of human-computer interaction, computer-supported collaborative learning, and learning analytics.



**PEDRO G. CAMPOS** received the B.S. and M.S. degrees in software engineering from the Universidad de Tarapacá, Arica, Chile, in 2006, and the M.S. and Ph.D. degrees in computer engineering from the Universidad Autónoma de Madrid, Madrid, Spain, in 2013. He is currently an Associate Professor at the Information Systems Department, Universidad del Bío-Bío, Concepción, Chile. His research interests include recommender systems, machine learning, and software engineering.

...